Feasibility restoration for iterative meta-heuristics search algorithms

Randall, Marcus

Published in:
Developments in Applied Artificial Intelligence - 15th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2002, Proceedings

DOI:
10.1007/3-540-48035-8_17

Licence:
Unspecified

Feasibility Restoration for Meta-heuristic Search Algorithms

Marcus Randall[1]

School of Information Technology
Bond University, QLD 4229

Abstract. Many combinatorial optimisation problems have constraints that are difficult for meta-heuristic search algorithms to process. One approach is that of feasibility restoration. This technique allows the feasibility of the constraints of a problem to be violated and then brought back to a feasible state. The advantage of this is that the search can proceed over infeasible regions, thus potentially exploring difficult to reach parts of the state space. In this paper, a generic feasibility restoration scheme is proposed for use with neighbourhood search algorithms. In addition, some comments are made on incorporating this technique into agent based meta-heuristics such as ant colony optimisation.

Keywords: Heuristic Search, Feasibility Restoration, Autonomous Agents.

1 Introduction

Iterative meta-heuristic search algorithms such as simulated annealing (SA) and tabu search (TS) are often implemented to solve combinatorial optimisation problems (COPs) using simple neighbourhood operators such as 2-opt, move and inversion [9]. In many cases, infeasible space is ignored and only transitions that maintain feasibility are used. However, traversing this space could lead to other regions in which higher quality solutions reside. One way to traverse this space is by the process of *feasibility restoration*. This technique allows the feasibility of the constraints of a problem to be violated and then brought back to a feasible state, thus overcoming infeasible regions.

Feasibility restoration algorithms have typically been designed for specific problems [1, 3–5, 7, 8]. A few examples will be detailed here. Chu and Beasley [4] use a heuristic to repair solutions generated by a genetic algorithm to solve generalised assignment problems (GAPs). It simply reassigns jobs from overloaded agents to agents that have some spare capacity. Kämpke [8] uses a similar approach in solving bin packing problems. After two items from different bins have been swapped, the capacity restriction of either of the bins may be violated. Therefore, another function is used to assign the largest item in the overfilled bin to the bin with the most spare capacity. Hertz, Laporte and Mittaz [7] use augmented local search operators on a variant of the vehicle routing problem

(VRP) known as the capacitated arc routing problem. They employ the PASTE operator which merges various vehicle routes. Often the feasibility (in the form of vehicle capacity) of the solution is violated so as such a new heuristic operator, known as SHORTEN, replaces portions of the route with shorter paths while still maintaining the required edge set.

Abramson, Dang and Krishnamoorthy [1] use an entirely different approach to any of the above. They use the general 0-1 integer linear programme (ILP) feasibility restoration method of Connolly [6] (explained in Sect. 2), but tailor it to suit the set partitioning problem.

The characterisation of the aforementioned feasibility restoration schemes is that they are all tailored to solve specific optimisation problems. As such, if a problem definition changes or a new problem is required to be solved, new feasibility restoration algorithms would need to be developed. This paper presents the opposite approach by investigating a new generic feasibility restoration algorithm and is organised as follows. Sect. 2 describes how feasibility restoration can be achieved in a generic manner. This is given in the context of a general representation system for COPs based on linked lists. Sect. 3 describes the further research using the developed feasibility restoration model.

2 Generic Feasibility Restoration

One of the first attempts at generic feasibility restoration was by Connolly [6] in a programme called GPSIMAN. GPSIMAN is a general purpose SA solver that accepts 0-1 ILPs. It incorporates a general mechanism for restoring feasibility of the system after each transition.

The feasibility restoration technique flips variables (other than the original variable changed by the SA process, called the *primary transition*) in order to obtain a new feasible solution. The scheme employed by Connolly [6] is a heuristic technique whereby a score is computed for each of the variables based on how helpful a change in the variable value would be for feasibility restoration. The most helpful variable (the one with the highest score) is flipped and the resulting feasibility/infeasibility is recalculated. If feasibility has been restored, the procedure is terminated. However, in many instances, particularly for 0-1 problems which have many related constraints, this is not the case. The algorithm proceeds to calculate the next most helpful variable. This progresses as a depth wise tree search, in which the algorithm can backtrack, should it find that the current sequence of flips cannot restore feasibility. This procedure is only useful if feasibility is relatively easy to restore, else the search for feasibility can quickly degenerate. If the process cannot restore feasibility after a fixed number of searches, then the primary transition is rejected.

According to Abramson and Randall [2], this algorithm is very slow and only effective for very small problems (such as a 5 city travelling salesman problem). In fact, they report that typically the restoration part of the algorithm takes 30% of the runtime. As such, in order for feasibility restoration to be usable, a new approach needs to be developed. The rest of this section outlines possible

algorithms based on the linked list representation system (explained next) and local neighbourhood search operators.

2.1 List Representation System

In order to develop a generic feasibility restoration scheme, it is helpful to express COPs in a uniform representation language. Randall and Abramson [9][1] use a language based on dynamic lists has been used to represent COPs for iterative meta-heuristics. The language directly represents a solution to a COP by using lists and sub-lists. As most COPs require an arrangement or grouping of elements to form a solution, different sub-lists can be used to represent different groups. Consider the GAP [4] for instance. Each sub-list can represent a particular agent and contain a collection of jobs for that agent to perform. As the solution changes (transitions are made), the lists shrink and grow accordingly. Therefore, problems represented by lists require two sets of constraints:

- *List Constraints* describe the list structure of the problem. E.g., for the GAP, the number of sub-lists (agents) must remain static and each sub-list must have a variable size with one being the lower limit (i.e. each agent must perform at least one job).
- *Problem Constraints* define the feasible space of the problem. E.g., for the GAP, the problem constraints represent the resource capacity of each agent.

The system has the advantage that common local search operators can be directly applied to the solution list. For instance, the move operator shifts one element from one sub-list to another.

2.2 Generic Feasibility Restoration Algorithm

As discussed in Sect. 1, the aim of feasibility restoration is to traverse infeasible space in order to find another feasible pocket of solutions that may be inaccessible otherwise. It is possible to apply feasibility restoration techniques to both the list constraints and problem constraints. In the version described in this paper, only the operation regarding the latter type is discussed.

The process of feasibility restoration consists of first breaking and then repairing feasibility. A transition is made that potentially violates some or all of the problem constraints. A series of feasibility maintaining/preserving transitions is subsequently undertaken in order to return to a feasible state.

As noted in Sect. 2, Connolly's [6] algorithm uses a feasibility restoration scheme based on backtracking. The central idea in backtracking approaches is that the primary transition must remain after the restoration process has been completed. If feasibility cannot be restored, then the primary transition is undone. The generic backtracking feasibility restoration algorithm that is applicable for local search transition operators is given in Fig. 1. Note, the amount of feasibility is calculated by comparing the left-hand side to the right-hand side of each constraint and considering the relational operator [9].

[1] The reader is referred to this reference for a complete description of the list representation system.

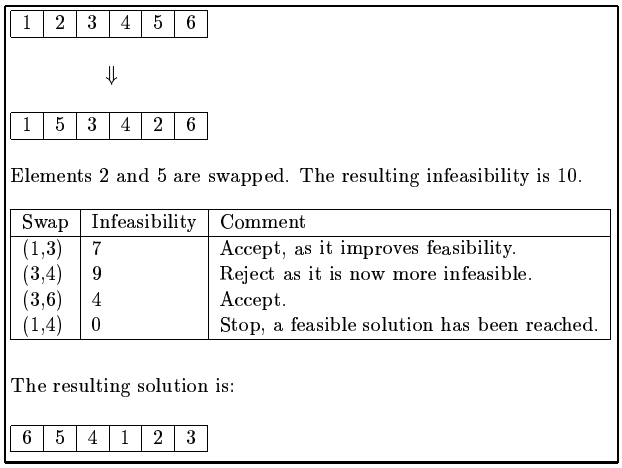
Fig. 1. A generic feasibility restoration algorithm.

```
X' = X;
Status = success;
Perform the primary transition(X');
C0 = Calculate infeasibility(X');
if (C0 > 0)
        Status = Restore_feasibility(X,C0,0);
End if;
if (Status == success) X = X';
End.

Function Restore_feasibility(X', C0, level)
Perform a transition that does not disturb the elements affected by
the primary transition(X');
C1 = Calculate infeasibility(X');
If (C1 > C0)
        Repeal the transition(X');
        Disallow this transition at this level;
        If (there are no more transitions at this level)
                level = level - 1;
                If (level == 0) return failed;
        End if;
        Return Restore_feasibility(X', C1, level);
Else
        If (C1 == 0)
                Return success;
        Else
                Return Restore_feasibility(X', C1, level + 1);
        End if;
End if;
End Resore_feasibility;

Where:
        X is the solution.
```

Consider the example in Fig. 2 that uses the swap operator as an application of the algorithm in Fig. 1. The example does not refer to a particular problem, though it could represent a single truck tour in a VRP under a tour length constraint.

Modified Algorithm The algorithm in Fig. 1, like its Connolly [6] counterpart, will potentially require much computational effort. Hence, a more efficient modified algorithm is presented in Fig. 3. The move operator will be used to demonstrate this approach (though it is applicable to other local search operators). In the context of this operator, the algorithm is useful for problems such

Fig. 2. Example of a backtracking feasibility restoration algorithm using the swap operator.



Elements 2 and 5 are swapped. The resulting infeasibility is 10.

Swap	Infeasibility	Comment
(1,3)	7	Accept, as it improves feasibility.
(3,4)	9	Reject as it is now more infeasible.
(3,6)	4	Accept.
(1,4)	0	Stop, a feasible solution has been reached.

The resulting solution is:

6	5	4	1	2	3

as the GAP and VRP in which elements can be moved between sub-lists. In this algorithm, a combination of elements is either removed from the sub-list or added to the sub-list from other sub-lists in order to make the constraints associated with the original sub-list feasible. For practical purposes, the number of elements that constitute the combination is bounded by a constant value. The search for a combination terminates when one that satisfies the problem constraints associated with the sub-list is found.

3 Further Research

This paper described the use of generic feasibility restoration techniques for meta-heuristic search algorithms such as SA and TS. At the present time, code is being developed and refined to perform generic feasibility restoration with the move local search operator. The next step in this development is to produce restoration techniques for all operators that follow the algorithm in Fig. 3.

Some preliminary investigation shows that optimal and near optimal solutions are gained using this feasibility restoration technique. However, a cursory comparison on runtimes with Randall and Abramson [9] reveals that this algorithm is slower at the moment. There is no question though that it is an improvement on Connolly's GPSIMAN. Work needs to be done in order to increase the efficiency of the restoration algorithm. One way would be to employ feasibility restoration operations less frequently. For instance, it could be used only to escape local optima.

The algorithms described in this paper each utilise one local search transition operator at a time. It may be possible to use a variety of operators and calculate which one is likely to restore feasibility the fastest. In addition, more research

Fig. 3. A modified feasibility restoration algorithm for the move operator.

```
Move an element to another sub-list;
Check the constraints associated with the old sub-list;
If (these constraints are violated)
        Attempt to add a combination of elements on this sub-list;
        If (this fails) try to do the same thing except remove the
        elements off the old sub-list;
        If (the old sub-list is now feasible) perform
        feasibility maintenance with the displaced elements;
        If (this fails) abort the restoration, reinstate the
        original solution and exit;
End if;
Check the constraints associated with the new sub-list;
If (these constraints are violated)
        Attempt to add a combination of elements from this list;
        if (the new sub-list is now feasible) perform
        feasibility maintenance with the displaced elements;
        If (this fails) try to add to this sub-list a combination of
        elements;
        If (this fails) abort the restoration, reinstate the
        original solution;
End If;
End.
```

needs to be undertaken to establish a cost/benefit comparison against feasibility maintaining operations. This needs to be carried out across a range of problems.

Incorporating feasibility restoration operators into constructive meta-heuristics, such as those that belong to ant colony optimisation (ACO), presents some challenges. According to Randall and Tonkes [10], constraints for constructive heuristics fall into two categories: those that can be satisfied throughout the search process and those that can only be satisfied with a complete solution. A feasibility restoration operator would potentially violate the former group of constraints. There are two possible ways to deal with this:

1. use a modified form of the algorithm in Fig. 3 that can cope with a partial solution or
2. remove elements from the solution (apart from the primary element) that now violate the constraints and allow the constructive technique to add back feasible elements. This is a form of backtracking that has not yet been investigated in the context ACO (i.e. ants have only been studied in a unidirectional sense).

References

1. Abramson, D., Dang, H. and Krishnamoorthy, M. (1996) "A Comparison of Two Methods for Solving 0-1 Integer Programs Using a General Purpose Simulated Annealing", Annals of Operations Research, 63, pp. 129-150.
2. Abramson, D. and Randall, M. (1999) "A Simulated Annealing Code for General Integer Linear Programs", Annals of Operations Research, 86, pp. 3-21.
3. Beasley, J. and Chu, P. (1996) "A Genetic Algorithm for the Set Covering Problem", European Journal of Operational Research, 94, pp. 392-404.
4. Chu, P. and Beasley, J. (1997) "A Genetic Algorithm for the Generalised Assignment Problem", Computers and Operations Research, 24, pp. 17-23.
5. Chu, P. and Beasley, J. (1998) "A Genetic Algorithm for the Multidimensional Knapsack Problem", Journal of Heuristics, 4, pp. 63-86.
6. Connolly, D. (1992) "General Purpose Simulated Annealing", Journal of the Operational Research Society, 43, pp. 495-505.
7. Hertz, A., Laporte, G. and Mittaz, M. (2000) "A Tabu Search Heuristic for the Capacitated Arc Routing Problem", Operations Research, 48, pp. 129-135.
8. Kämpke, T. (1988) "Simulated Annealing: A New Tool in Bin Packing", Annals of Operations Research, 16 pp. 327-332.
9. Randall, M. and Abramson, D. (2001) "A General Meta-heuristic Based Solver for Combinatorial Optimisation Problems", Journal of Computational Optimization and Applications, 20, pp. 185-210.
10. Randall, M. and Tonkes, E. (2001) "Solving Network Synthesis Problems using Ant Colony Optimisation", Lecture Notes in Artificial Intelligence, 2070, Springer Verlag: Berlin, pp. 1-10.